ESTIMATION OF LATERAL-DIRECTIONAL PARAMETERS FROM FLIGHT DATA USING NEURAL NETWORKS

by

SUNIL KHUBCHANDANI

DEPARTMENT OF AEROSPACE ENGINEERING

ESTIMATION OF LATERAL-DIRECTIONAL PARAMETERS FROM FLIGHT DATA USING NEURAL NETWORKS

A Thesis Submitted
In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

Sunil Khubchandani

to the

DEPARTMENT OF AEROSPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

ACKNOWLEDGEMENT

I express my deep sense of gratitude to my esteemed teachers and thesis supervisors, Dr S.C. Raisinghani and Prof. A.K. Ghosh for their invaluable guidance, constructive criticism and persistent encouragement throughout this work. I shall always remain indebted to them for the precious time they have spared for me and the patience with which they always enlightened my path.

With all sincerity, I thank Dr. P.K. Kalra, the professor of Electrical Engg. Deptt. for his guidance and all the facilities he provided me for finishing my work in time.

I have no words to express my thanks to my parents and my youngers who have been constant source of inspiration to me. I wish to thank all my friends and well wishers who made my stay at I.I.T.K. memorable and pleasant.

SUNIL G K.

DEDICATED TO

MY FAMILY

MR. G.K. KHUBCHANDANI

MRS. KAVITA DEVI KHUBCHANDANI

VAISHALI, NIDHI AND HITESH

Abstract

Application of Feed Forward Neural Network (FFNN)) to Aerospace Engineering problem is one of the current topics of interest. The FFNNs do not require an a priori model of aircraft for modeling, and provide a black - box type of model of the aircraft by mapping suitable network inputs to chosen network outputs. The present thesis deals with modeling of aircraft lateral-directional dynamics through FFNNs and application of recently proposed Delta-method for the estimation of aircraft stability and control derivatives (parameters). For modeling, aircraft motion variables and the control inputs are used as the input file while aerodynamic force or moment coefficients are used as the output file for training FFNN. For the purpose of parameter estimation, the trained FFNN is now presented with suitably modified input file and the corresponding predicted output files of force and moment coefficients are obtained. Suitable interpretation and processing of these input-output files results in estimated values of the parameters. The method is verified on lateral-directional simulated flight data. A detailed study has been carried out to show how the accuracy of estimates gets affected when flight data corresponding to various types and forms of aileron and/or rudder control inputs are analyzed. It is shown that the combination of aileron and rudder applied in certain sequential form leads to better estimates. Finally, it is also shown that the Delta-method is quite robust and can be used for parameter estimation even if flight data contains measurement noise

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

SYMBOLS

C_l	rolling moment coefficient
C_n	yawing moment coefficient
C_y	force coefficient along Y-axis
E	error cost function
I_{xx}	moment of inertia about x-axis kg-m^2
S	reference area m^2
U_1	velocity, m/sec
b	span, m
f	nonlinear sigmoidal activation function
m	aircraft mass, kg
n_i	number of nodes in input layer
n_h	number of nodes in hidden layer
n_o	number of nodes in output layer
p	roll rate, rad/sec
r	yaw rate, rad/sec
s	half the span, m
w	weight matrix between two layers
y	estimated output
z	desired output
β	sideslip angle, deg or rad
δa	aileron deflection, rad
δr	rudder deflection, rad
λ	logistic gain or slope factor
μ	learning rate parameter

Ω momentum rate parameter

Superscript

$\cdot$ derivative with respect to time

T transpose of matrix

Stability And Control Derivatives

$C_{lp} = \partial C_l / \partial (pb/2U_1)$, $C_{np} = \partial C_n / \partial (pb/2U_1)$, $C_{yp} = \partial C_y / \partial (pb/2U_1)$,

$C_{lr} = \partial C_l / \partial (rb/2U_1)$, $C_{nr} = \partial C_n / \partial (rb/2U_1)$, $C_{yr} = \partial C_y / \partial (rb/2U_1)$,

$C_{l\beta} = \partial C_l / \partial \beta$, $C_{n\beta} = \partial C_n / \partial \beta$, $C_{y\beta} = \partial C_y / \partial \beta$,

$C_{l\delta a} = \partial C_l / \partial \delta a$, $C_{n\delta a} = \partial C_n / \partial \delta a$, $C_{y\delta a} = \partial C_y / \partial \delta a$,

$C_{l\delta r} = \partial C_l / \partial \delta r$, $C_{n\delta r} = \partial C_n / \partial \delta r$, $C_{y\delta r} = \partial C_y / \partial \delta r$

INTRODUCTION

The process of estimating numerical values of aerodynamic stability and control derivatives (parameters) from flight test data is termed as Parameter Estimation. The mathematical model representing airplane dynamics requires accurate values of these parameters. Such mathematical models are useful for verification of stability augmentation systems (SAS) and in-flight simulators.

Three distinct approaches for estimating these stability and control derivatives are

(i) Theoretical Methods

(ii) Wind Tunnel Testing

(iii) Flight Testing

At initial design stages of aircraft design, theoretical methods[1-3] provide the only convenient way of estimating the aircraft parameters. However the accuracy of such theoretical estimates being not so high, there is need to verify these estimates with those obtained from wind tunnel testing and flight testing. Wind tunnel methods although improve accuracy of estimation of parameters, they are time consuming and expensive. Further, simulation of control surfaces, power effects and stringent flight conditions are difficult to simulate satisfactorily. Wind tunnel estimates also suffer from discrepancies due to interference effects of support system, wall effects, turbulence level etc. It is, therefore, desirable that the wind tunnel estimates be corroborated with estimates from actual flight test data.

Three popular methods to estimate stability and control derivatives from flight test data are as follows :

(i) Equation Error Method

(ii) Output Error Method

(iii) Maximum Likelihood Method

The principle of least squares is used in Equation Error Method. The error gets minimized with respect to unknown parameters in each of the equations. Its advantages include computational simplicity, non-iterative nature and applicability to both linear as well as nonlinear models. However this method can not be applied if all the states are not measured accurately and gives poor results if measurements are noisy. Thus in order to get accurate results ample efforts are required for data reconstruction and smoothing. Some times these data processing tasks are more complicated than the task of parameter estimation itself. The errors between the measured and model response produced by identical inputs is minimized by Output Error Method. The method assumes that there exist no modeling errors. The method processes the measured noise while assuming the model representation of given system to be exact. A comprehensive survey of these methods is reported by Maine and Iliff[4]. The methods like analog matching[5], Newton-Raphson Method[6], Modified Newton-Raphson Method[6] etc. fall in this category.

Maximum likelihood[7] estimates are those for which the observed value would be the most likely to occur. The main advantage of this method is that estimated parameters are asymptotically

approximation which can collectively model any nonlinear relationship between the inputs and the outputs

ANN can only operate with given input-output data. Some networks require only input variables and are called as auto associative networks while others require input-output pairs and are called as hetro associative networks.

Nodes or neurons are the sites where all the computations are performed. Figure 1 illustrates the most common type of neuron. Each neuron collects information from abutting connections and produces single output value. ANNs also consist of edges(connections) between the nodes which not only allow information to flow in particular defined direction but also assign weight to the information. Such connection weights are termed as synaptic weights. These weights are adjusted in accordance to a well defined algorithm which helps to capture knowledge within the network

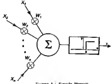

Figure 1 : Simple Neuron

Due to the above properties, ANNs are widely preferred for mapping any nonlinear relationship between the input and output. It can be stated that it is a "black box model structure" because functional relationship between the input and output is not explicitly known.

There are several types of ANNs, but only two of them have especially found application in the field of aircraft system identification. These are -

(i) Feed Forward Neural Network (FFNN)

(ii) Recurrent Neural Networks (RNN)

FFNNs lead to a black box model in which no physical significance can be added to either network structure or to the network weights[12-14]. RNNs have fixed structure and possess fixed number of nodes,equal to number of unknown parameters.

FFNNs have neurons arranged in layers like directed graphs and thus are static in nature. RNNs are dynamic neural networks incorporating an output feedback.In recent paper Raol and Jategaonkar[15] have shown amenability of RNNs to state space modeling and thereby demonstrated their applicability to explicitly estimate aircraft parameters. However as pointed by authors,[15] R NNs have only a limited scope for aircraft identification applications and it is the FFNNs which may prove to be more flexible and thereby have higher potential for future applications for modeling aircraft dynamics and aircraft parameter estimation.

Recently, system identification has been re-examined by many

researchers using ANNs[12-15]. Significant contributions in this direction have been made by Hess[12], Yossef and Juang[13], Basappa and Jategaonkar[14], Raol and Jategaonkar[15], and Linse and Stengal[16]. Hess[12] dealt with use of FFNNs to represent aircraft aerodynamics. Yossef et al[13] demonstrated the feasibility of neural modeling approach to establish a nonlinear aerodynamic model that is amenable to flight test data processing. Basappa and Jategaonkar[15] have studied various aspects of FFNN modeling and its applicability to real flight data. Raol and Jategaonkar[15] have shown use of RNN for aircraft parameter estimation. Linse and Stengal[16] have shown accurate modeling of aerodynamic coefficients using a system identification model composed of an extended Kalman-Bucy filter for state and force estimation and a computational neural network for aerodynamic model.

In a recent paper Raisinghani et al[17] have proposed two new methods for estimation of airplane parameters using FFNNs. The proposed methods are called Delta-Method and Zero-Method. Both methods have been validated on simulated flight data generated for longitudinal dynamics of an example aircraft. Out of these two Delta Method was shown to be more reliable, consistent and robust. We have therefore chosen Delta method for present study wherein its applicability to extract lateral-directional stability and control parameters from flight data is explored. Specifically following aspects of application of FFNN to aircraft lateral-directional dynamics are dealt with:

(i) Aerodynamic modeling of lateral-directional dynamics using

FFNN wherein force and moment coefficient(C_l, C_n, C_y) are mapped to measured motion variables p, r, β and control inputs $\delta a, \delta r$. Various aspects about identifying suitable FFNN structure and associated network parameters like learning rate, momentum rate, slope etc. are studied.

(ii) Application of Delta-method for estimating lateral-directional parameters from simulated flight data for an example aircraft is demonstrated. A detailed study has been carried out to show how accuracy of estimates is affected by the type of control input used for generating flight data. Results are also presented to indicate the effect of measurement noise present in the simulated flight data on the accuracy of estimates.

The layout of the present work is as follows.

After introduction in chapter one, details of FFNNs and backpropagation algorithm are given in chapter two. Chapter three deals with the method used for parameter estimation and various aspects about the generation of simulated flight data. Chapter four consists of descriptive study of the results obtained. Lastly, chapter five briefs about the conclusions drawn on the basis of results obtained and also about the recommendations for future work.

CHAPTER 2

FEED FORWARD NEURAL NETWORKS

Feed Forward Neural Networks

2.1 General

FFNNs are multi layered perceptrons consisting of an input layer an output layer and one or more hidden layers. The number of neurons(nodes) in the input layer and the output layer are determined respectively by the number of input and output variables while the number of neurons in the the hidden layer(s) is decided by the complexity of the problem. Figure 2 shows a general schematic picture of FFNN with two hidden layers consisting of many neurons.

FFNNs are static and characterized by unidirectional flow of variables i.e. the inputs are propagated through the hidden layers to the output layer. Each node in the layer(input or hidden) is connected to each node in the next layer(either the hidden or the output) through a connective weight. Every node in the input layer and the hidden(s) layer is biased. The nodes in the hidden layer(s) and the output layer have a nonlinear activation function, e.g. sigmoidal or hyperbolic tangent function. The hidden nodes in FFNN have good approximation capability and thereby allow the network to build a model of arbitrary complexity between the input(s) and the output(s).

For specific case of aircraft lateral-directional dynamics the input variables to the network are the motion variables p, r, β and the aircraft control input signals δa and/or δr while the output variable is aerodynamic force or moment coefficient i.e.

yawing moment coefficient(C_n), or rolling moment coefficient(C_l), or force coefficient along y-axis(C_y). Figure 3 shows body fixed axes system. The known input output data is first used to train the network. The predicted output values of C_l,C_n,C_y are compared with the corresponding known values of C_l or C_n or C_y and the errors are backpropagated using a method called the back propagation algorithm (BPA). This process results in updating of network connective weights. The training of the network is continued till the mean square error (MSE), defined as follows, is less then the prescribed value.

Mean square error (MSE) defined by equation (1) is used to test the convergence. The influence parameters are varied until the MSE is within the defined limits. Mean square error can be expressed as

$$MSE = \frac{1}{Nm} \sum_{k=1}^{N} \sum_{j=1}^{m} [\, z_j(k) - u_j(k) \,]^2 \qquad (1)$$

where, N is the number of data points , m is the number of input variables , z is the desired or measured response and u is the estimated response of the network.

The details of BPA and forward propagation in FFNN are described next.

2.2 Backpropagation Algorithm (BPA)

Backpropagation[8] is a simple iterative gradient descent algorithm. The essential idea behind the BPA , which is widely

used for training multi layered perceptrons , is to view the error function as a function of network weights or parameters and to perform gradient descent in the weight parameter space to search for a minimum error between desired and estimated value . In this method , weights are updated incrementally so that , at each iteration a small step is taken in the negative direction of the error to make it small more rapidly

There are two common types of BPN learning algorithms

Batch or Sweep BPN

Sequential or Recursive BPN

The batch BPN updates the network weights after presentation of the complete training data set . Hence , a training iteration incorporates one sweep through all the training patterns

In case of recursive BPA , also called pattern learning , the network weights updated sequentially as the training data set is presented . The recursive BPA is more convenient and efficient as compared to the batch BPA .

A brief summary of the BPN algorithm for one hidden layered FFNN is given below for completeness

2.2.1 Initialization of Weights

The input and output weight matrices of the network are initialized randomly . With reference to figure 2 the order of weight matrices and weight vector is as follows :

Matrix/Vector	Symbol	Order of matrix/vector
(i) weight matrix between input and hidden layers	w_1	$n_h * n_i$
(ii) bias weight vector	w_{10}	$n_h * 1$
(iii) weight matrix between hidden and output layer	w_2	$n_0 * n_h$
(iv) bias weight vector	w_{20}	$n_0 * 1$

where n_i , n_h and n_0 are the number of nodes in the input hidden and output layers respectively

2 2 2 Forward propagation

During the forward propagation , for a given input vector u_0 , the estimated output vector u_2 is computed

(i) Propagation from input layer to hidden layer

$$y_1 = w_1 * u_0 + w_{10} \qquad (2a)$$

$$u_1 = f (y_1) \qquad (2b)$$

where y_1 is the vector of intermediate variables , u_1 is the vector node output at the hidden layer and f is the vector of nonlinear sigmoidal node activation function , defining the node characteristic

It should be noted that each component of the vector on left hand side corresponds to the respective component of the vector on right hand side for all the equations .

The sigmoidal function may be written as

$$f_i(y_i) = \frac{1}{1 + e^{-(y_1(i)/\lambda)}} \qquad i=1,2, \ldots ,n_h \qquad (3)$$

where λ is the logistic gain or slope factor of the hidden layer activation function . All the nodes at any particular layer have generally the same slope factor

(ii) Propagation from hidden layers to output layer

$$y_2 = w_2 * u_1 + w_{20} \qquad (4a)$$

$$u_2 = f(y_2) \qquad (4b)$$

where y_2 is the vector of intermediate variables , u_2 is the vector node output at the output layer and f is the vector of nonlinear sigmoidal node activation function

Similarly , the sigmoidal function in this case is

$$f_i(y_2) = \frac{1}{1 + e^{-(y_2(i)/\lambda)}} \qquad i=1,2,\ldots,n_0 \qquad (5)$$

2.2.3 Recursive Learning Algorithm

The backpropagation learning algorithm is based on optimizing a suitably defined error cost function . At each point the local output error cost function , which is sum of squared errors , is given by

$$E_k = \frac{1}{2}(z - u_2)^T(z - u_2) \qquad (6)$$

where k is the discrete time data index , z is the measured response vector , u_2 is the network estimated response vector and $e_2 = (z - u_2)$ denotes the error .

Minimization of eq (6), applying the steepest gradient descent method , also called delta rule , yields

$$w_2^{k+1} = w_2^k + \mu * (- \partial E_k / \partial w_2) \qquad (7)$$

where $\mu > 0$ is learning rate parameter determining the speed of convergence, $(\partial E_k / \partial w_2)$ is the gradient of error cost function with respect to w_2. A judicious choice of learning rate parameter, μ, is necessary to ensure reasonable convergence rate.

Partial differentiation of eq.(6) with respect to the elements of weight matrix w_2 and substitution of Eqs. (4a & 4b) yields gradient of the error cost function

$$(\partial E_k / \partial w_2) = - f' (y_2) (z - u_2) u_2^T \qquad (8)$$

where $f' (y_2)$ is the derivation of the output node activation function i.e. eq. (5)

Defining e_{2b} as,

$$e_{2b} = f' (y_2) (z - u_2) \qquad (8a)$$

and by adding momentum term to Eq (7), the weight update rule for the output layer is obtained as

$$w_2^{k+1} = w_2^k + \mu \, e_{2b} \, u_2^T + \Omega (w_2^k - w_2^{k-1}) \qquad (9)$$

where Ω is momentum term.

Over relaxation of the gradient update i.e. of eq. (7), through the last term on the right hand side of eq. (9), called momentum term, helps to damp out the parasitic oscillations in the weight update, e.g., due to large value of learning rate, μ. Approximately, it amounts to increasing the learning rate from μ to $\mu / (1 - \Omega)$ without magnifying the parasitic oscillations.

Similarly, the partial differentiation of eq. (6) with respect to w_1 and substitution of eq (2 a & 2 b) yields

$$(\partial E_k / \partial w_2) = - f' (y_1) \, w_2^T \, e_{2b} \, u_0^T \qquad (10)$$

where $f'(y_1)$ is the derivative of hidden layer activation function i. e. of eq (3)

$$e_{1b} = f'(y_1)\, w_2^T\, e_{2b} \qquad (10a)$$

The weight update rule for the hidden layer is given by

$$w_1^{k+1} = w_1^k + \mu\, e_{1b}\, u_0^T + \Omega\,(w_1^k - w_1^{k-1}) \qquad (11)$$

For the sigmoidal function in eq (3)and eq(5) chosen respectively as the hidden and output layer node activation function respectively, the derivatives $f'(y_1)$ and $f'(y_2)$ are given by

$$f'(y_1) = \frac{e^{-(y(l)/\lambda)}}{\lambda\,(1 + e^{-(y(l)/\lambda)})^2} \qquad (12)$$

It may be pointed out that the training algorithm is recursive, as stated above, the weights are updated sequentially for each time point using eqs.(9 & 11)

From computational view point, the backpropagation algorithm (BPA) requires

(i) Computation of the hidden & output node activation function i.e. eqs (2a & 2b) and eqs (4a & 4b)

(ii) Computation of the error cost function i. e. eqs (8a & 10a)

(iv) Computation of the derivatives of the node activation function (s Eq (12) for hidden and output layer

(v) Computation of the new weights from Eqs. (9 & 11) until the desired accuracy is obtained

2.3 Modeling, Prediction And Influence Parameters of FFNN

As stated earlier problem solving using FFNN consist of two

phases namely (i) learning or training phase and (ii) prediction phase

In modeling, learning or training phase procedure consists of presenting the known input and output data to network and adjusting the network weights using backpropagation algorithm (BPA). During this procedure network captures the functional relationship existing between the inputs and outputs (not necessarily in the same form as in the original physical model) in form of network weights. Once modeling or training is over and network weights and topology is frozen, the same data is passed through the network with the same topology to check for prediction capability of the network based on some convergence criteria (MSE in the present study).

There are several network parameters which affect the convergence rate. These influence parameters are:

1. Number of Hidden Layers (H L)
2. Number of neurons in each hidden layer (nn's)
3. Learning Rate Parameter (μ)
4. Momentum Rate Parameter (Ω)
5. Number of iteration (I)
6. Initial Network Weights (R S.)
7. Weight Limit
8. Logistic Gain or slope Factor (λ)

Determination of optimal topology is a cumbersome job. It is hit and trial method which is like part science and part art. In standard literature on FFNN, some thumb rules have been proposed

for structure and parameters of FFNN's for various applications One has to still develop one's own set of thumb rules by repeated trials of training the network for the problem at hand. We have found and recommended a set of values of influence parameters for mapping lateral-directional dynamics of an aircraft. These values are presented in chapter four

CHAPTER 3

PARAMETER ESTIMATION METHOD

AND

SIMULATED FLIGHT DATA GENERATION

PARAMETER ESTIMATION METHOD

AND

SIMULATED FLIGHT DATA GENERATION

3.1 General

In this chapter we first describe the salient features of Delta-method[17] used for parameter estimation from flight data using Feed Forward Neural Networks. Next we outline the procedure used and types of simulated data generated for analysis via Delta-method. Details of the example airplane used and types of control inputs utilized for generating simulated flight data are also given. Flight data with and without pseudo measurement noise were prepared for analysis. Although real flight data for lateral-directional dynamics of an airplane were recently obtained and are being separately analyzed by others, we did use the control input form used for real flight data to generate corresponding simulated flight data for analysis and comparison with the results obtained by others. Scarcity of time did not permit analysis of the complete real flight data to be included in the present work.

3.2 Delta-method

Parameters occurring in the equations of motion of an aircraft represent partial derivatives of aerodynamic force and moment coefficients with respect to the corresponding motion or control input variables. In other words these parameters can be thought of as variation in aerodynamic coefficients due to small

variation in one of the motion or control variable about the nominal value in such a way that only that particular variable is allowed to change while rest of the variables are held constant at their nominal values. This fundamental understanding is exploited in Delta-method for estimating aircraft parameters.

Let us now assume that FFNN is already trained to map the network input variables $p, r, \beta, \delta a$ and δr to the output variables C_l or C_n or C_y. Now only one of the network inputs is given a small(delta,Δ) perturbation at each time point while all others are held at their original values. Such a modified input file is now presented to the trained neural network to predict the perturbed value of aerodynamic coefficients at its output node. The difference in the predicted value of aerodynamic coefficient from the value predicted for the original values of inputs is due to the perturbation in the value of the chosen network input variable. The difference so calculated in the value of aerodynamic coefficient divided by the perturbation value yields the corresponding aircraft parameter. For example say β is perturbed by small value $\Delta\beta$ and difference observed in the predicted value of C_l for β and $\beta+\Delta\beta$ is ΔC_l. Then $\Delta C_l/\Delta\beta$ yields the stability derivative $C_{l\beta}$. Similarly perturbation in δa and corresponding ΔC_n observed will yield control derivative $C_{n\delta a}$ where $C_{n\delta a} = \Delta C_n/\Delta\delta a$.

To avoid bias due to one sided differences, motion and control variables are perturbed in both increasing and decreasing direction. For example C_y values for $\beta+\Delta\beta$ and $\beta-\Delta\beta$ are predicted

to be C_y^+ and C_y^- respectively, then $C_{y\beta} = C_y^+ - C_y^- / 2\Delta\beta$

It should be noted that for each of the perturbed value at each time point there will be a corresponding predicted value of aerodynamic coefficient, and hence a different calculated value of the parameter estimate. Ideally all such values should have been identical to yield a single value of the parameter. However in reality there is variation in predicted values. Histograms of all the estimated values showed a near normal distribution. Thus the mean value is used as the estimate and standard deviation about the mean as the measure of accuracy of the estimates.

3.3 Simulated Data Generation

In the absence of real flight data, simulated data is used as the input-output data to train the FFNN. This simulated data is obtained by solving the lateral-directional equations of motion for an example aircraft. For the present study, the example aircraft was chosen to be the DLR research aircraft ATTAS, details of which are given in the DLR report by Jategaonkar[16]. This aircraft was chosen as real flight data for it was acquired recently and are being analyzed by others. Thus it would help comparison with results from real flight data at a later date. Also, we could utilize the control input form used for real flight and we have analyzed the corresponding simulated data for parameter estimation. For generation of simulated data equation of motion used are given next.

3.4 Equations Of Motion

For generation of lateral-directional simulated flight data, following perturbed equations of motion[18] were used

$$v = -r\,U_1 + p\,w + g\cos\theta\,\sin\phi + (\bar{q}S/m)\,C_y \qquad (13a)$$

$$w = -p\,v + q\,U_1 + g\cos\theta\,\cos\phi + (\bar{q}S/m)\,C_z \qquad (13b)$$

$$p = [1.0/(I_x I_z - I_{xz}^2)]\{\bar{q}Sb(I_z C_l + I_{xz} C_n) - qr(I_{xz}^2 + I_z^2 - I_y I_z) + pq(I_x - I_y + I_z)\} \qquad (13c)$$

$$r = [1.0/(I_x I_z - I_{xz}^2)]\{\bar{q}Sb(I_x C_n + I_{yz} C_l) + qp(I_{xz}^2 + I_x^2 - I_y I_x) - rq I_{yz}(I_x - I_y + I_z)\} \qquad (13c)$$

$$\phi = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \qquad (13e)$$

$$\theta = q\cos\phi - r\sin\phi \qquad (13f)$$

where,

$$\beta = v/U_1\ ;\ \bar{q} = (\rho\,U_1^2)/2.0$$

$$C_l = C_{lp}(pb/2U_1) + C_{lr}(rb/2U_1) + C_{l\beta}\beta + C_{l\delta a}\delta a + C_{l\delta r}\delta r \qquad (14a)$$

$$C_n = C_{np}(pb/2U_1) + C_{nr}(rb/2U_1) + C_{n\beta}\beta + C_{n\delta a}\delta a + C_{n\delta r}\delta r \qquad (14b)$$

$$C_y = C_{yp}(pb/2U_1) + C_{yr}(rb/2U_1) + C_{y\beta}\beta + C_{y\delta a}\delta a + C_{y\delta r}\delta r \qquad (14c)$$

The steady flight condition(about which the perturbations were initiated), mass, moment of inertia, and geometric measures for the aircraft, taken from reference[18], are as follows :

$U_1 = 103.0$ m-sec^{-1}

$M = 16352.23$ N

$g = 9.81$ m-sec^{-2}

$\rho = 0.7$ kg-m^{-3}

$S = 64$ m^2

$c = 3.159$ m

$b = 21.5$ m

$I_{xx} = 162312.2$ kg-m^2 $\quad$ $I_{yy} = 252687.0$ kg-m^2

$I_{zz} = 388440.97$ kg-m^2 $\quad$ $I_{xz} = 11442.0$ kg m^2

where U_j is the velocity ρ is density of cruising altitude of aircraft under consideration. b is the span, $\bar{c}$ is mean aerodynamic chord. S is reference area and I's are the moment of inertia about the mentioned axes.

Fourth order Runge-Kutta method is used to solve above sets of equations for generating motion variables p(t),r(t),β(t) for some known control input form of δa(t) and δr(t). The network input file is prepared using these values of p,r,β,δa and δr. The force and moment coefficient $C_l(t), C_n(t), C_y(t)$ are calculated from eqs (14a-14c) and these form network output variable(s). For simulated flight data, true values of the parameters given in Table 1 were fed into the above set of equations (14a-14c).

CHAPTER 4
RESULTS AND DISCUSSION

Results And Discussion

Simulated flight data for Lateral-Directional dynamics of an aircraft were generated for different combinations of aileron and rudder control inputs. The network input file consisted of motion variables (p, r, β) along with aircraft control inputs (δa and/or δr). The network output file was either C_l or C_n or C_y. The network was trained using such input-output files for many such sets of simulated flight data. The network tuning parameters were varied to obtain the optimal architecture that was used for parameter estimation for all the sets of flight data. To this purpose network parameters were varied in the range shown below. The final values selected for the present study are also shown below in the third column.

Network Parameter	Range of variation tried	Final Values
Logistic Gain	0.65-1.0	0.85
Random Seed	0.5-0.6	0.6
Number of Hidden Layers	1-3	1
Number of Nodes in Hidden Layer	4-10	6
Number of Iteration	2000-15000	5000
Learning Rate	0.3-0.6	0.3
Momentum Rate	0.4-0.8	0.5

The criterion for selection of network parameters was based on MSE observed at the end of selected number of iterations. It was found that MSE decreased up to about 5000 iterations, and beyond it, the decrease was marginal. To save on computational

time, for all the further studies the architecture along with number of iterations was kept fixed at 5000. Although it is possible that slight variations in the network parameters may result in lower MSE for specific set of flight data it was decided to freeze the architecture of network for all studies reported herein due to limitation of time. Since present work is concerned with parameter estimation via Delta-method, we refrain from reporting detailed discussion on training phase and effects of network parameters on the MSE and thereby on the match obtained between the desired and the predicted values of C_l or C_n or C_y. The effect of network parameters on training was similar to that reported in the literature, i.e., Basappa and Jategaonkar[14] and Rakesh Kumar[19].

As reported in literature, multistep 3-2-1-1 type of control inputs have been found very efficient for generation of flight data to be used for parameter estimation. A multistep 3-2-1-1 type of input consists of control deflection for 3,2,1 and 1 second in +ve,-ve,+ve and -ve direction respectively. The maximum magnitude was kept at 0.1 or 0.05 radians, and duration was seven seconds. To bring out the relative merits of 3-2-1-1 type of input, it was of interest to explore parameter estimation using flight data for other types of control inputs. Two such control inputs, namely an arbitrary and a sinusoidal (Fig.4) were used for flight data generation. For such a comparative study following five types of control inputs were used to obtain corresponding flight data for parameter estimation via. Delta-method

case I : Multistep 3-2-1-1 aileron input, maximum amplitude 0.1 radians and duration seven seconds.

case II : Multistep 3-2-1-1 rudder input, maximum amplitude 0.1 radians and duration seven seconds.

case III : An arbitrarily varying aileron control input with maximum magnitude 0.1 radians and duration seven seconds.

case IV : A sinusoidal aileron input of maximum amplitude 0.1 radians and duration seven seconds.

case V : Combination of identical multistep 3-2-1-1 aileron and rudder control inputs. Both having maximum amplitude as 0.1 radians and duration seven seconds.

Using these different inputs, the simulated flight data was generated and parameters were predicted via Delta-method. The results corresponding to case I – V are listed respectively in columns 3 – 7 of Table 1, along with true values of parameters in column 2 for ready comparison.

case I (Table 1, column 3) : As seen from Table 1, except for C_{yp} and C_{yr} all the other parameters are well estimated. However, values of standard deviation for some of the parameters are on higher side. Further, control derivatives like $C_{l\delta r}$, $C_{n\delta r}$, $C_{y\delta r}$ cannot be estimated since there is no rudder input available in the network input file for perturbation as required by Delta-method.

case II (Table 1, column 4) : Results for δr alone show degradation as compared to those for δa alone. Comparison with 3rd

column (case II) shows that, in addition to C_{yp}, C_{yr} as in case I, C_{np}, C_{nr} and $C_{y\beta}$ are also poorly estimated. Again, control derivatives for aileron ($C_{l\delta a}$, $C_{n\delta a}$, $C_{y\delta a}$) can not be estimated for this case, since no δa is available in network input file.

case III (Table 1, column 5) : Compared to case I, the arbitrary δa control input results showed detoriation in estimated values, specially in C_{lr}, C_{np} and C_{nr} values while C_{yp} and C_{yr} estimates remained poor as in case I. Thus multistep 3-2-1-1 input signal shows its superiority over the arbitrary input signal.

case IV (Table 1, column 6) : The parameter estimates were poor except for β-derivatives ($C_{l\beta}$, $C_{n\beta}$, $C_{y\beta}$) as compared to the cases discussed above. This is due to the fixed frequency of the sinusoidal signal which is unable to excite all the modes of the aircraft dynamics.

case V (Table 1, column 7) : All the above cases had only one of the controls (δa or δr) used for exciting aircraft dynamics, and thereby we could estimate only the corresponding control derivatives, i.e., either δa or δr. Simultaneous, identical δa and δr inputs were used for generating flight data and so control derivatives for both δa and δr could be estimated by perturbing them in network input file. Estimates from this data showed that like previous cases, C_{yp} and C_{yr} were still being poorly estimated and estimates for both δa and δr ($C_{l\delta a}$, $C_{n\delta a}$, $C_{y\delta a}$, $C_{l\delta r}$, $C_{n\delta r}$, $C_{y\delta r}$) were also poor. A close scrutiny of the estimated values of these control derivatives showed that there was a pair-wise high correlation between ($C_{l\delta a}$, $C_{l\delta r}$) .

$(C_{y\delta a}, C_{y\delta r})$ and $(C_{n\delta a}, C_{n\delta r})$ in the sense that both the control derivatives of the pair had almost equal values. Initially this result looked very strange but on reflection, it was clear that indeed the values should be either exactly equal or almost equal as was the case. The reason for expecting such equal estimates for both is that we are showing the neural network two identical inputs namely, δa and δr, but they are indistinguishable from neural network point of view. It is like showing the same input twice to neural network.

The above observation required a relook at the type of control inputs to be used for generating flight data so as to enable estimation of all the stability and control derivatives with better accuracy. As a first step we used the real control inputs that has been used in generating real flight data provided by DLR Germany. We could not analyze the whole set of this real flight data as these were received very late when the thesis work was nearing completion. Analysis of complete data set is being carried out by another Ph.D. candidate.

From the data set supplied we chose two types of control inputs forms which are put under case VI and case VII.

case VI : Pulse type aileron input (Fig.5)

case VII : Combination multistep 3-2-1-1 type aileron and rudder inputs (Fig.6a and 6b)

Using these control inputs, simulated data was generated and parameters estimated via Delta-method. The results for these two cases are given in table 2 and discussed below.

case VI (Table 2 column 3) In general, the estimated parameters compared well with the true values except that the rudder control derivatives still could not be estimated and derivatives like C_{yp}, C_{yr} and C_{np} were not well estimated

case VII (Table 2, column 4) Estimated parameters in this case showed remarkable improvement in their estimated values when compared with case V Here not only were both control derivatives estimated separately but also their estimated values were close to the true values and standard deviation were reasonably low On the whole all the parameter estimates showed better comparison with the true values

On the face of it, such improved results were surprising and it was first suspected that this could be due to measurement noise that may be present in the control inputs copied out from the real flight data Alternatively, it was conjectured that this may be due to the rudder and aileron inputs being not identical as was the case for case V Thus, the real control inputs may have contained measurement noise and would not have identical shape To test this conjecture, flight data was generated for δa and δr combination where both δa and δr inputs had certain pseudo measurement noise added to them Specifically pseudo measurement noise of 1% and 5% was added to the control inputs that were used in case V i.e. combination of multistep 3-2-1-1 δa and δr The estimated parameters from such flight data are discussed below

case VIII (Table 2, column 5) Combination of multistep 3-2-1-1 δa and δr with 1% measurement

noise in δa and δr

The estimated parameters did not show any improvement over that estimated in case V. The estimated values of pair-wise control derivatives still showed trend similar to the one mentioned under case V.

case IX (Table 2, column 6) : Combination of multistep 3-2-1-1 δa and δr with 5% measurement noise in δa and δr

Estimated parameters still followed the same trend in their estimated values as in case VIII.

At this stage, difficulty of estimating control derivatives when combination of δa and δr is used, led to idea of introducing delay between the two control inputs. As mentioned earlier, identical control inputs δa and δr would confuse the neural network and it would not be able to distinguish between δa and δr inputs. It was hoped that the delay would give a chance to neural network to distinguish between these two control inputs i.e. δa and δr. Based on this conjecture, following types of inputs were used for data generation and analysis, results of which are given in table 3.

case X (Table 3, column 3) : Combination of multistep 3-2-1-1 δa and δr with rudder control delayed by 1.0 second i.e. rudder input was initiated 1 second after aileron input is applied

The estimated parameters clearly reflected the success of

using delay between the two inputs. Not only the control derivatives got resolved but also were very close to their true values [$C_{y\delta a}$ being only exception] and with reasonably low standard deviation. However, stability derivatives C_{nr}, C_{yr}, C_{yp} and C_{lp} were still on higher side. It seemed desirable to explore and find control input form that might reduce these standard deviations.

case XI [Table 3, column 4] : Combination of multistep 3-2-1-1 δa and δr with aileron control delayed by 1.0 second.

Like case X, parameters were again well estimated except for some deterioration in $C_{y\delta a}$ and C_{lr} values.

case XII [Table 3, column 5] : Combination of multistep 3-2-1-1 δa and δr with rudder control delayed by 2.0 second.

Estimated parameters from this study yielded good estimated values for all derivatives except C_{nr}, C_{yp} and C_{yr}. Overall the 2.0 second delay was not as impressive as 1.0 second delay.

In the light of the potential shown by time delay between the applications of the two control inputs, the idea was further extended wherein second control followed at the end of the first control. The following two cases were investigated towards this goal.

case XIII [Table 4, column 3] : Combination of multistep 3-2-1-1 δa and δr such that aileron input was applied for the first seven

	seconds, rudder input kept zero and for the next seven seconds rudder input being applied keeping aileron input zero
case XIV (table 4, column 4)	Combination of multistep 3-2-1-1 δa and δr such that rudder input is applied for the first seven seconds and aileron input for the next seven seconds.

The estimated parameters from case XIII and XIV, clearly indicate that for such type of control inputs, their exists no confusion for neural network. It can very easily distinguish between two inputs. It was also observed that in case XIV the standard deviations were much lower. Also the above type of application of control inputs resulted in good estimates of even the weak derivatives like $C_{yp}, C_{yr}, C_{np}, C_{lp}, C_{nr}$ and $C_{y\delta a}$

For illustration,specifically for case XIII, the actual, the trained and the predicted values of C_l, C_n, C_y are plotted in Fig 7a, 7b, and 7c. By actual value we mean those values of force or moment coefficients which were calculated using eq (14) with true values of parameters for particular control input of case XIII.As values of these force or moment coefficients were present as output in the input-output file used to train the FFNN, the values so obtained after training the FFNN for optimal network parameters are termed as trained values. The predicted values are those values which are obtained by using of the estimated parameters in eq (14). It may be seen from Fig. 7 that the actual,

trained and predicted C_l, C_n and C_y show good match and thus indicate validation of Delta method for predicting parameter estimates for flight data.

Also histograms for all the estimated parameters are plotted in fig. 8a, 8b and 8c. These figures clearly indicate the near normal distribution for most of the estimated parameters and justify the concept of mean values and standard deviation for these estimated parameters.

Since the real flight data is generally noisy in nature the effect of measurement noise on the estimated parameters was carried out to show how the accuracy of the estimates is affected by presence of measurement noise. For this purpose pseudo measurement noise was added to p,r,β and $C_l$or $C_n$or C_y corresponding to simulated flight data of case X. The results for it are given under case XV and case XVI below :

case XV (table 4, column 5) : case X + 1% measurement noise in

p,r,β and $C_l$or $C_n$or C_y

case XVI (table 4, column 6) case X + 5% measurement noise in

p,r,β and $C_l$or $C_n$or C_y

In case XV most of the parameters are well estimated as was observed for case X (no noise). This indicates that presence of low measurement noise does not affect the accuracy of estimates and this reflects on the robustness of Delta-method. A comparison of results for 1% (case XV) and 5% (case XVI) cases shows the increase in noise level does result in marginally poorer estimates for most of the parameters. Curiously, some of the weak derivatives like C_{yp}, C_{yr} and $C_{y\delta a}$ are actually better estimated

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In the present work Delta-method using FFNNs has been used for estimation of lateral-directional parameters from simulated flight data. Different types of control inputs were used to generate corresponding sets of flight data for analysis through Delta-method. Comparison of parameter estimates so obtained showed how the accuracy of estimates depends on the types of control input used for flight data. In particular, it was observed that a time delay between aileron and rudder inputs leads to better estimates of aileron and rudder control derivatives. Further, multistep 3-2-1-1 type aileron and rudder lead to better excitation of aircraft dynamics and thereby to better estimates as compared to pulse, sinusoidal or any arbitrary varying input. The Delta-method has been found to be robust enough to provide good parameter estimates even in presence of measurement noise in the flight data.

5.2 Recommendations

It would be of great interest to validate Delta-method on real flight data. In particular, if data could be generated for different types of control inputs studied in the present work, then these could be readily compared to the results reported herein. The idea of time delay between aileron and rudder inputs is worth verifying using real flight data. In present work, force and moment coefficients have been mapped separately, it would be

interesting to investigate where C_l C_n and C_Y are mapped together and then apply Delta method for parameter estimation.

REFERENCES

1. Seckel, E., and Morris, J. J "The Stability Derivatives of the Navion Aircraft Estimated by Various Methods and Derived from Flight Test Data," FAA-RD-71-6

2. Seckel, E., "Stability and Control of Airplanes and Helicopters," Academic Press, New York, 1964

3. Ellison, D E , "USAF Stability and Control Hand Book (DATCOM)," Wright Patterson Airforce Base, OHIO, Revised August 1968

4. Maine, R E and Iliff K W.,"Identification of Dynamic System - Theory and Formulation," NASA RP 1138 Feb 1985

5. Platrass, A "Determination of Aerodynamic Derivatives of the FIAT G9 IT3 Aircraft from Flight Test by means of Manual Analog Model Matching" Paper No N75-19257 of Publication ESRO TT-104, Nov. 1974

6. Taylor, L W., Iliff, K W., and Powers, B. G "A comparison of Newton-Raphson and other Methods for Determining Stability Derivatives from Flight Data," AIAA Paper No. 69-315, May 1969

7. Klein, V ,"Identification Evaluation Methods "AGARD-LS-104, Nov 1979

8. Widrow, B ,"30 Years of Adaptive Neural Networks Perceptrons, Madaline and Backpropagation,"Proceedings of the IEEE, Vol 78, No 9, Sep. 1990, pp 1415-1441

9. Chen, S , Cowan, C. F N ,"Parallel Recursive Prediction

Error Algorithm for Training Layered Networks,"Int. J Control, 1990, Vol 50, No. 6, pp. 1225-1228

10 Chen, S Billings, S. A. and Grant, P M ,"Nonlinear System Identification Using Neural Networks,"Int J Control, 1990, Vol 51 No 6, pp. 1191-1214

11 Sjoberg, J. Hjalmarsson H. and Ljung, L.,"Neural Networks in System Identification,"Proceedings of the 10th IFAC Symposium on System Identification, 4-6 July 1994 Copenhagen, Denmark, Vol 2, pp. 49-71

12. Hess, R A ,"On the Use of Backpropagation with Feed Forward Neural Networks for the Aerodynamic Estimation Problem," AIAA-93-3639-CP 1993

13 Youseff, H M ,"Estimation of Aerodynamic Coefficients using Neural Networks," AIAA paper 93-3639, Aug 1993.

14. Basappa and Jategaonkar, R. V.,"Aspects of Feed Forward Neural Network Modelling and its Application to Lateral - Directional Flight Data," DLR IB 111-95/30, Sep 1995

15 Raol J R and Jategaonkar R V.,"Aircraft Parameter Estimation Using Recurrent Neural Networks - A critical appraisal," AIAA Paper 95-3504-c, Aug. 1995

16 Linse, D J, and Stengel, R F," Identification of Aerodynamic Coefficients Using Computational Neural Networks," Journal of Guidance Control and Dynamics, Vol 16, No 6, Nov -Dec 1993, pp. 1018-1025

17 Raisinghani S.C., Ghosh A.K and Kalra P K "Two New Techniques For Aircraft Parameter Estimation Using Neural

Networks"(submitted for publication)

18 Jategaokar R "Identification Of The Aerodynamic Model Of The DLR Reasearch Aircraft ATTAS From Flight Test Data" DLR -FB90-40, Sept. 1990

19 Rakesh Kumar , "Parameter Estimation From Flight Data Using Feed Forward Neural Networks" M Tech Thesis Aerospace Engg Department, I I T -Kanpur, February '96

TABLE 1

Parameter	True Value	Estimated Parameters case1	case2	case3	case4	case5
$-C_{lp}$	0.98	1.1	0.77	0.80	0.50	0.92
		(0.18)	(0.13)	(0.16)	(0.18)	(0.15)
C_{lr}	0.42	0.40	0.44	0.64	-0.17	0.58
		(0.13)	(0.07)	(0.12)	(0.24)	(0.20)
$-C_{l\beta}$	0.126	0.095	0.101	0.162	0.10	0.11
		(0.025)	(0.02)	(0.03)	(0.034)	(0.03)
$-C_{l\delta a}$	0.23	0.26	NE	0.22	0.12	0.096
		(0.05)		(0.04)	(0.05)	(0.03)
$C_{l\delta r}$	0.046	NE	0.066	NE	NE	-0.90
			(0.012)			(0.025)
$-C_{np}$	0.115	0.14	0.86	0.07	0.06	0.10
		(0.04)	(0.20)	(0.05)	(0.03)	(0.09)
$-C_{nr}$	0.495	0.66	0.77	0.12	0.34	0.60
		(0.17)	(0.13)	(0.18)	(0.22)	(0.19)
$C_{n\beta}$	0.28	0.25	0.20	0.19	0.21	0.26
		(0.06)	(0.05)	(0.07)	(0.06)	(0.05)
$C_{n\delta a}$	0.0	0.001	NE	-0.004	0.003	-0.077
		(0.003)		(0.006)	(0.006)	(0.03)
$-C_{n\delta r}$	0.16	NE	0.16	NE	NE	-0.08
			(0.03)			(0.03)

C_{yp}	0.30	0.42	3.93	0.19	0.47	−0.003
		(0.19)	(1.3)	(0.21)	(0.17)	(0.04)
C_{yr}	0.727	0.53	2.01	−0.20	0.70	0.27
		(1.0)	(0.43)	(1.0)	(0.9)	(0.85)
$-C_{y\beta}$	1.132	0.95	0.72	0.84	0.95	1.1
		(0.26)	(0.16)	(0.30)	(0.30)	(0.25)
$C_{y\delta a}$	0.03	0.036	NE	0.024	0.07	0.095
		(0.01)		(0.03)	(0.03)	(0.05)
$C_{y\delta r}$	0.19	NE	0.16	NE	NE	0.12
			(0.03)			(0.06)

@ = standard deviation

NE = not estimated

TABLE 2

Parameter	True Value	Estimated Parameters			
		case6	case7	case8	case9
$-C_{lp}$	0.98	1.1	0.91	0.92	0.92
		(0.21)[@]	(0.11)	(0.15)	(0.15)
C_{lr}	0.42	0.42	0.41	0.58	0.58
		(0.07)	(0.12)	(0.19)	(0.20)
$-C_{l\beta}$	0.126	0.084	0.121	0.11	0.11
		(0.03)	(0.03)	(0.03)	(0.03)
$-C_{l\delta a}$	0.23	0.25	0.253	0.095	0.08
		(0.06)	(0.04)	(0.03)	(0.03)
$C_{l\delta r}$	0.046	NE	0.07	0.089	0.09
			(0.01)	(0.025)	(0.026)
$-C_{np}$	0.115	0.18	0.05	0.10	0.10
		(0.08)	(0.11)	(0.09)	(0.08)
$-C_{nr}$	0.495	0.55	0.64	0.60	0.60
		(0.18)	(0.19)	(0.18)	(0.17)
$C_{n\beta}$	0.28	0.27	0.28	0.26	0.26
		(0.08)	(0.05)	(0.05)	(0.05)
$C_{n\delta a}$	0.0	-0.001	0.023	-0.077	-0.073
		(0.003)	(0.026)	(0.03)	(0.025)
$-C_{n\delta r}$	0.16	NE	0.18	-0.078	-0.082
			(0.04)	(0.03)	(0.024)

C_{yp}	0 30	0 36	0 24	−0 003	−0 005
		(0 18)	(0 33)	(0 04)	(0 04)
C_{yr}	0 727	( )3	0 20	0 27	0 28
		(0 6)	(0 8)	(0 85)	(0 86)
$-C_{y\beta}$	( )32	0 97	(07	))	) (
		(0 29)	(0 23)	(0 25)	(0 25)
$C_{y\delta a}$	0 03	0 05	0 028	0 095	0 095
		(0 02)	(0 04)	(0 05)	(0 05)
$C_{y\delta r}$	0 19	NE	0 15	0 12	0 12
			(0 09)	(0 06)	0 06)

@ = standard deviation

NE = not estimated

C_{yp}	0.30	0.36	0.24	0.003	0.005
		(0.18)	(0.33)	(0.04)	(0.04)
C_{yr}	0.727	1.13	0.20	0.27	0.28
		(0.6)	(0.8)	(0.85)	(0.86)
$-C_{y\beta}$	1.132	0.97	1.07	1.1	1.1
		(0.29)	(0.23)	(0.25)	(0.25)
$C_{y\delta a}$	0.03	0.05	0.028	0.095	0.095
		(0.02)	(0.04)	(0.05)	(0.05)
$C_{y\delta r}$	0.19	NE	0.15	0.12	0.12
			(0.09)	(0.06)	0.06)

@ = standard deviation

NE = not estimated

TABLE 3

Parameter	True Value	Estimated Parameters		
		case10	case11	case12
$-C_{l p}$	0.98	1.09	1.03	0.995
		(0.18)[@]	(0.14)	(0.11)
$C_{l r}$	0.42	0.43	-0.30	0.37
		(0.08)	(0.13)	(0.07)
$-C_{l \beta}$	0.126	0.13	0.12	0.12
		(0.03)	(0.02)	(0.016)
$-C_{l \delta a}$	0.23	0.25	0.26	0.24
		(0.04)	(0.04)	(0.03)
$C_{l \delta r}$	0.046	0.04	0.043	0.042
		(0.012)	(0.014)	(0.016)
$-C_{n p}$	0.115	0.10	-0.043	0.08
		(0.04)	(0.08)	(0.08)
$-C_{n r}$	0.495	0.50	0.22	0.60
		(0.15)	(0.23)	(0.21)
$C_{n \beta}$	0.28	0.26	0.30	0.27
		(0.04)	(0.06)	(0.06)
$C_{n \delta a}$	0.0	0.004	0.004	-0.009
		(0.023)	(0.034)	(0.016)
$-C_{n \delta r}$	0.16	0.16	0.18	0.20
		(0.03)	(0.043)	(0.05)

C_{yp}	0.30	0.25	0.025	0.165
		(0.37)	(0.38)	(0.17)
C_{yr}	0.727	0.73	-0.34	0.53
		(0.22)	(0.87)	(0.4)
$-C_{y\beta}$	1.132	1.04	1.09	1.03
		(0.17)	(0.26)	(0.17)
$C_{y\delta a}$	0.03	0.011	0.004	0.054
		(0.06)	(0.025)	(0.07)
$C_{y\delta r}$	0.19	0.18	0.22	0.243
		(0.05)	(0.1)	(0.08)

@ = standard deviation

TABLE 4

Parameter	True Value	Estimated Parameters			
		case13	case14	case15	case16
$-C_{l_p}$	0.98	1.03	0.98	1.08	1.01
		(0.1)[@]	(0.03)	(0.17)	(0.21)
C_{l_r}	0.42	0.38	0.42	0.44	0.51
		(0.08)	(0.02)	(0.08)	(0.20)
$-C_{l_\beta}$	0.126	0.13	0.13	0.13	0.1
		(0.02)	(0.004)	(0.03)	(0.04)
$-C_{l_{\delta a}}$	0.23	0.26	0.25	0.25	0.23
		(0.03)	(0.007)	(0.04)	(0.04)
$C_{l_{\delta r}}$	0.046	0.045	0.047	0.04	0.05
		(0.001)	(0.003)	(0.013)	(0.02)
$-C_{n_p}$	0.115	0.12	0.12	0.101	0.095
		(0.07)	(0.04)	(0.14)	(0.13)
$-C_{n_r}$	0.495	0.54	0.52	0.49	0.47
		(0.06)	(0.11)	(0.14)	(0.14)
C_{n_β}	0.28	0.28	0.27	0.26	0.27
		(0.03)	(0.05)	(0.04)	(0.04)
$C_{n_{\delta a}}$	0.0	-0.001	0.02	0.005	0.006
		(0.001)	(0.05)	(0.024)	(0.02)
$-C_{n_{\delta r}}$	0.16	0.17	0.19	0.16	0.16
		(0.018)	(0.06)	(0.03)	(0.03)

C_{yp}	0.30	0.34	0.36	0.24	0.35
		(0.16)	(0.15)	(0.37)	(0.32)
C_{yr}	0.727	0.83	0.68	0.69	0.79
		(0.11)	(0.13)	(0.23)	(0.22)
$-C_{y\beta}$	1.132	1.11	1.11	1.05	1.02
		(0.09)	(0.1)	(0.17)	(0.17)
$C_{y\delta a}$	0.03	0.027	0.023	0.014	0.02
		(0.024)	(0.026)	(0.06)	(0.06)
$C_{y\delta r}$	0.19	0.21	0.19	0.18	0.19
		(0.03)	(0.02)	(0.05)	(0.05)

( ) = standard deviation

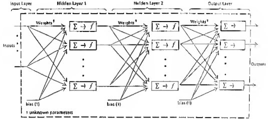

Fig. 2 Feed forward neural network with two hidden layers

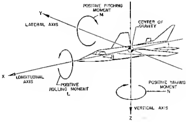

FIG 3 Aircraft Axes System

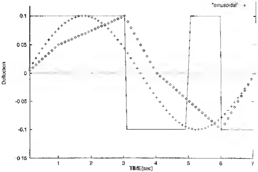

Fig. 4 Three Types Of Control Input Forms

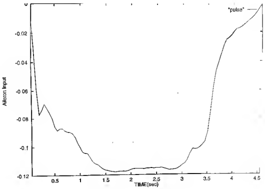

Fig 5 Pulse type control input (case VI)

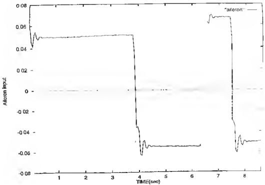

FIG. 6A multistep 3-2-1-1 aileron input signal (case VII)

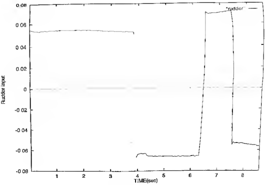

Fig. 6B multistep 3-2-1-1 rudder input signal (case VII)

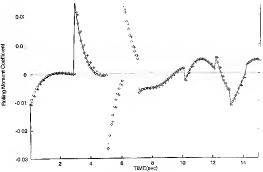

FIG 7A Comparison Of Actual, Trained and Estimated Response Of Rolling Moment Coefficient (C_l)

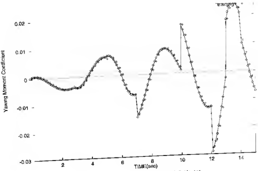

FIG. 7B Comparison Of Actual, Trained and Estimated Response Of Yawing Moment Coefficient (C_n)

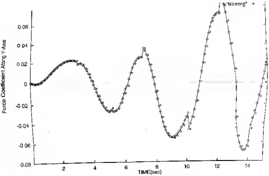

FIG 7c Comparison Of Actual, Trained and Estimated Response Of Force Coefficient Along Y axis (C_Y)

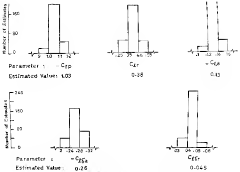

Fig. 8 a : Histograms for Estimated Parameters C_{lp}, C_{lr}, $C_{l\beta}$, $C_{l\delta a}$ and $C_{l\delta r}$

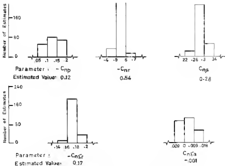

Fig. 8b Histograms for Estimated Parameters C_{np}, C_{nr}, $C_{n\beta}$, $C_{n\delta r}$, $C_{n\delta a}$

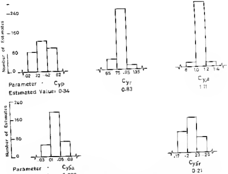

Fig 8c Histograms for Estimated Parameters C_{yp}, C_{yr}, $C_{y\beta}$, $C_{y\delta a}$ and $C_{y\delta r}$.